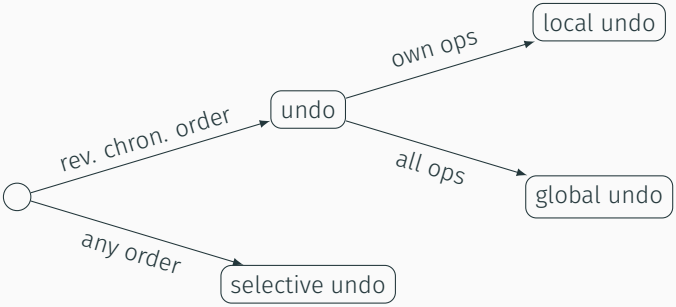
Extending Automerge: Undo and Redo

Leo Stewen, Martin Kleppmann

October, 2023

Undo Kinds in a Collaborative Setting



Illustration

Register: [■]

Figure 1: Local vs global undo.

Illustration

Register: [■]

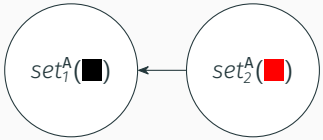


Figure 1: Local vs global undo.

Illustration

Register: [■]

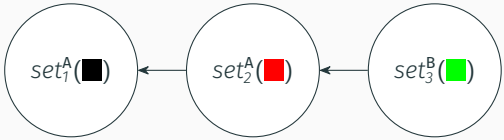


Figure 1: Local vs global undo.

Illustration

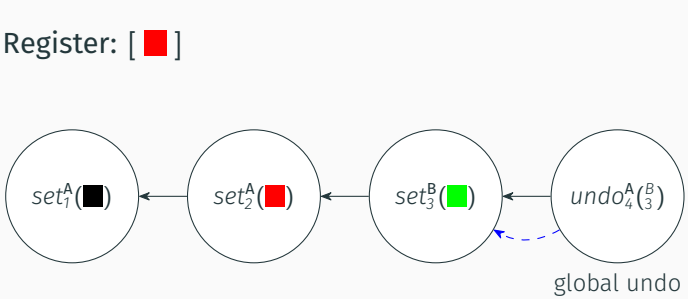


Figure 1: Local vs global undo.

Illustration

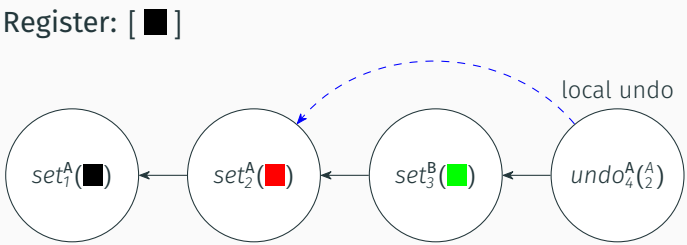


Figure 1: Local vs global undo.

Illustration

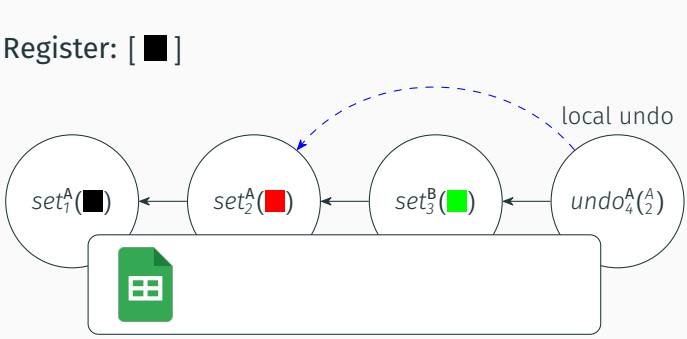


Figure 1: Local vs global undo.

Illustration

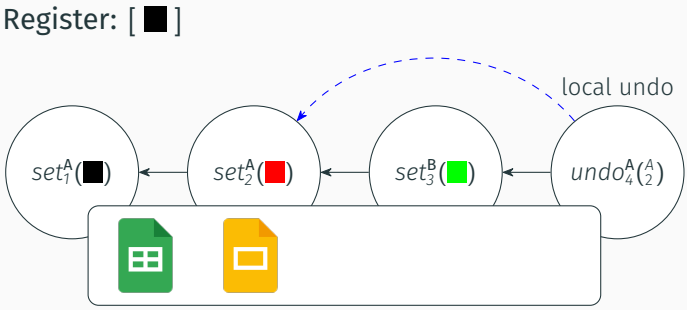


Figure 1: Local vs global undo.

Illustration

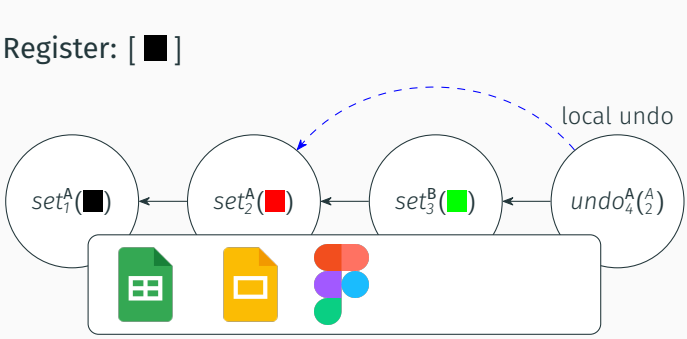


Figure 1: Local vs global undo.

Illustration

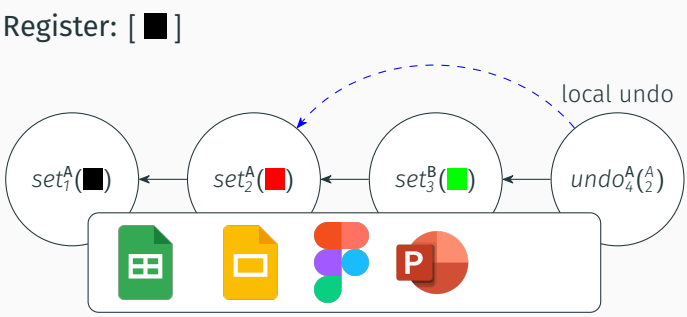


Figure 1: Local vs global undo.

Illustration

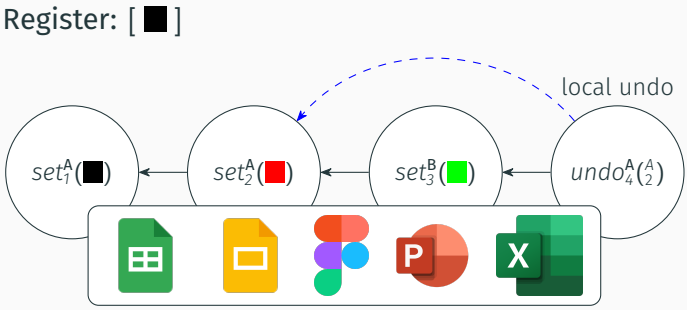
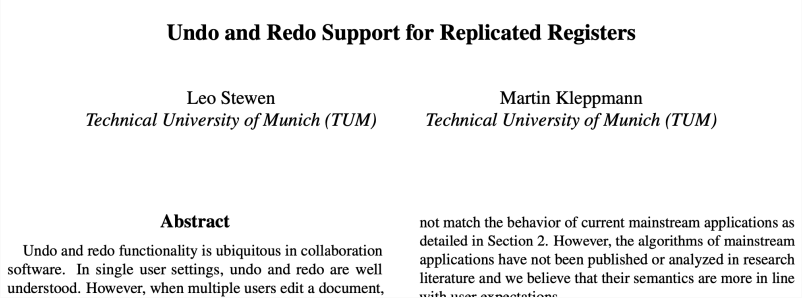


Figure 1: Local vs global undo.

Why Replicated Registers, if Automerge is a JSON CRDT?



Undo and Redo Support for Replicated Registers

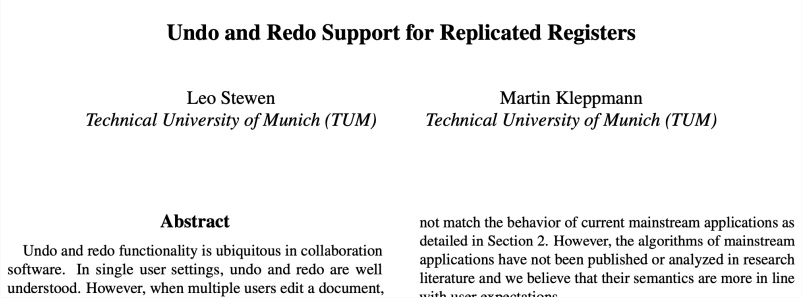
Leo Stewen
Technical University of Munich (TUM)

Martin Kleppmann
Technical University of Munich (TUM)

Abstract

Undo and redo functionality is ubiquitous in collaboration software. In single user settings, undo and redo are well understood. However, when multiple users edit a document, not match the behavior of current mainstream applications as detailed in Section 2. However, the algorithms of mainstream applications have not been published or analyzed in research literature and we believe that their semantics are more in line with user expectations

Why Replicated Registers, if Automerge is a JSON CRDT?



Undo and Redo Support for Replicated Registers

Leo Stewen
Technical University of Munich (TUM)

Martin Kleppmann
Technical University of Munich (TUM)

Abstract

Undo and redo functionality is ubiquitous in collaboration software. In single user settings, undo and redo are well understood. However, when multiple users edit a document, not match the behavior of current mainstream applications as detailed in Section 2. However, the algorithms of mainstream applications have not been published or analyzed in research literature and we believe that their semantics are more in line with user expectations

```
{
  A: a,
  B: [b1, b2]
}
```

Why Replicated Registers, if Automerge is a JSON CRDT?

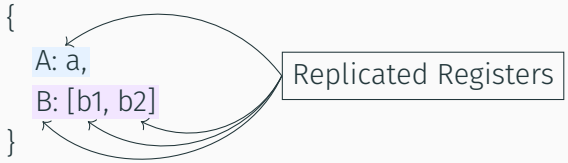
Undo and Redo Support for Replicated Registers

Leo Stewen
Technical University of Munich (TUM)

Martin Kleppmann
Technical University of Munich (TUM)

Abstract

Undo and redo functionality is ubiquitous in collaboration software. In single user settings, undo and redo are well understood. However, when multiple users edit a document, not match the behavior of current mainstream applications as detailed in Section 2. However, the algorithms of mainstream applications have not been published or analyzed in research literature and we believe that their semantics are more in line with user expectations



How are Replicated Registers different from, e.g., a Counter CRDT?

- every state update message contains the whole state

How are Replicated Registers different from, e.g., a Counter CRDT?

- every state update message contains the whole state
- what is the inverse of a set op?

How are Replicated Registers different from, e.g., a Counter CRDT?

- every state update message contains the whole state
- what is the inverse of a set op?
- how to redo?

Edge Cases to Consider (1)

Register: [■]

Figure 2: Undo of a merge op.

Edge Cases to Consider (1)

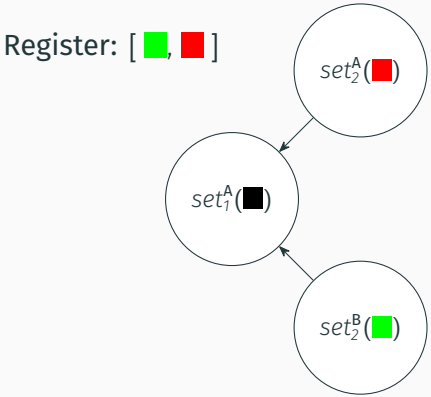


Figure 2: Undo of a merge op.

Edge Cases to Consider (1)

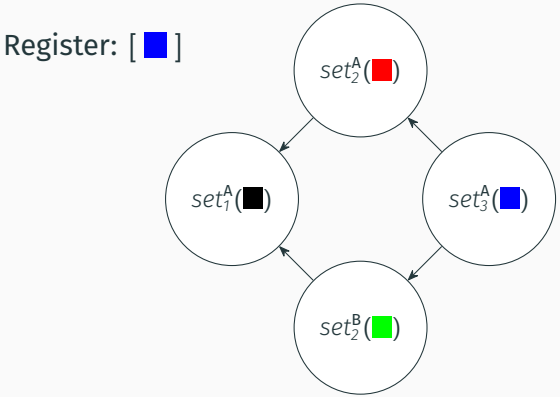


Figure 2: Undo of a merge op.

Edge Cases to Consider (1)

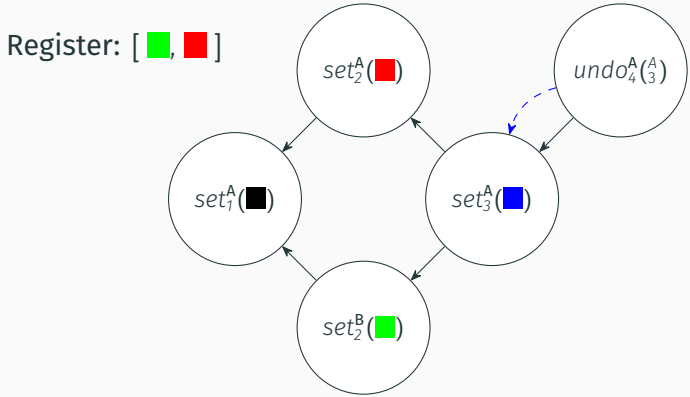


Figure 2: Undo of a merge op.

Edge Cases to Consider (2)

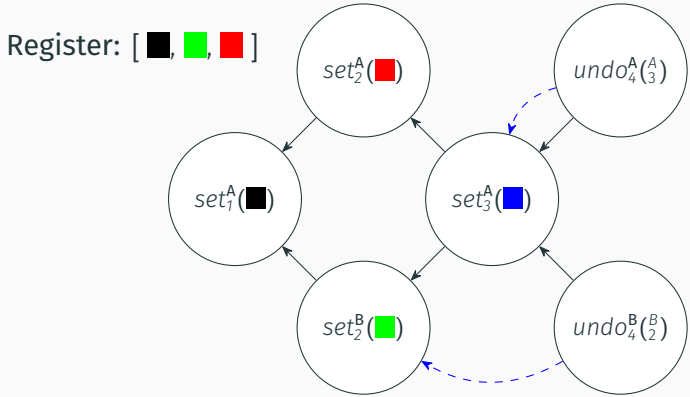


Figure 3: Concurrent undo ops.

Edge Cases to Consider (3)

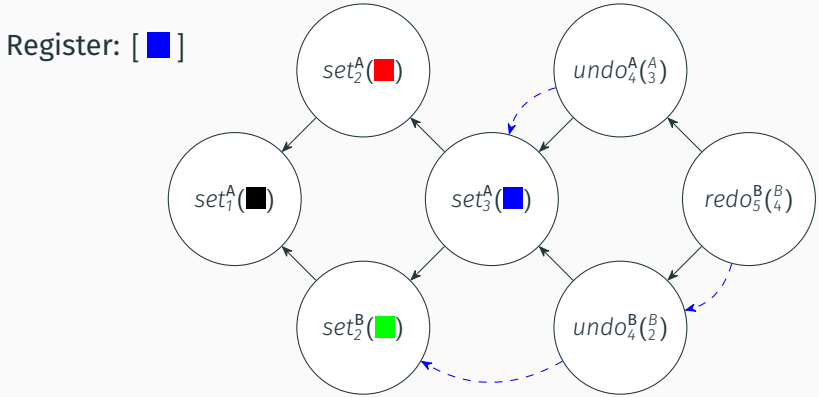


Figure 4: Redo restores state prior to its corresponding undo.

Edge Cases to Consider (4)

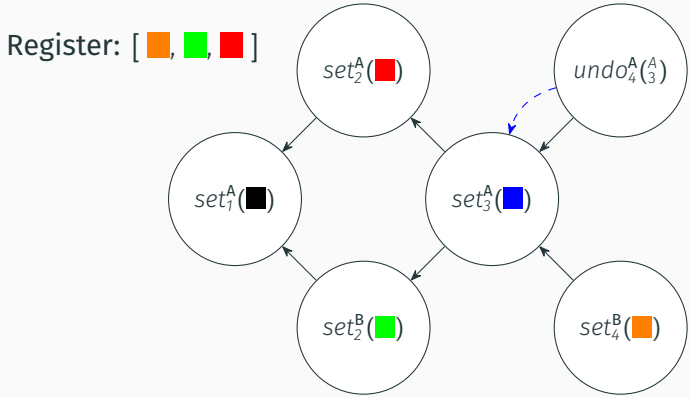


Figure 5: Concurrent undo and set op.

Constant Runtime for Common Editing Scenarios

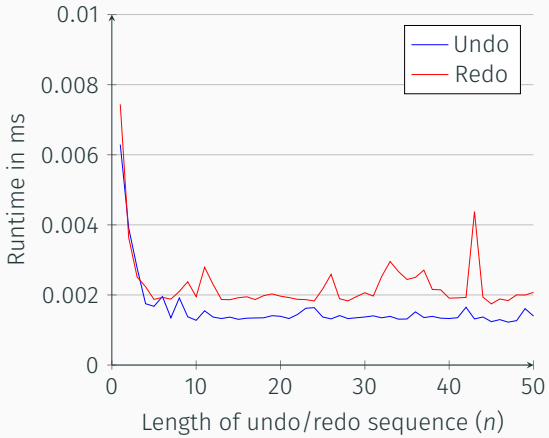


Figure 6: Runtime of the last undo/redo operation in a sequence of n consecutive undo/redo operations.

Questions? Feedback?

Reach us at
lstwn@mailbox.org
liangrun.da@tum.de
martin@kleppmann.com
LINK TO PAPERS! (as QR code?)

References

[1] W. Yu, L. André, and C.-L. Ignat, **“A CRDT supporting selective undo for collaborative text editing,”** in *15th IFIP WG 6.1 International Conference on Distributed Applications and Interoperable Systems*, ser. DAIS 2015, Springer LNCS volume 9038, Jun. 2015, pp. 193–206. DOI: 10.1007/978-3-319-19129-4_16.

Backup Slides

Figure 7: Canvas with two replicated registers.

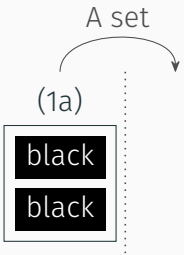


Figure 7: Canvas with two replicated registers.

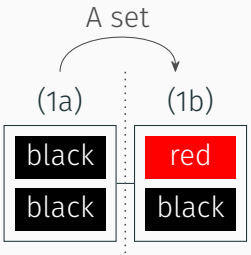


Figure 7: Canvas with two replicated registers.

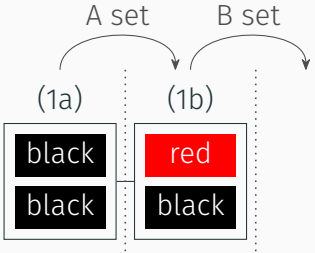


Figure 7: Canvas with two replicated registers.

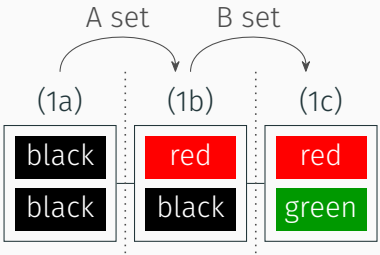


Figure 7: Canvas with two replicated registers.

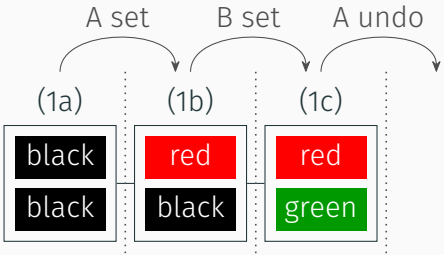


Figure 7: Canvas with two replicated registers.

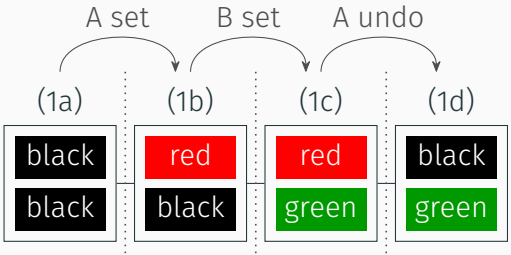


Figure 7: Canvas with two replicated registers.

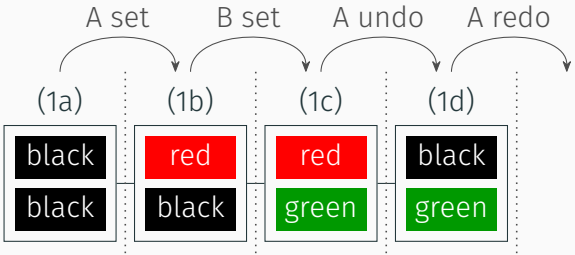


Figure 7: Canvas with two replicated registers.

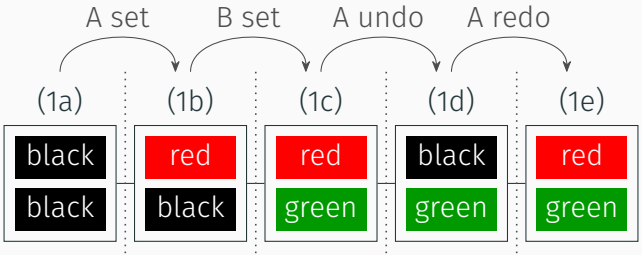


Figure 7: Canvas with two replicated registers.

Figure 8: Canvas with one replicated register.

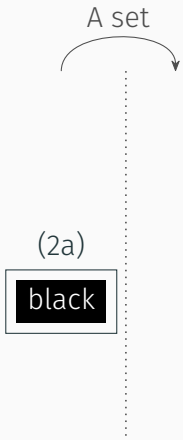


Figure 8: Canvas with one replicated register.

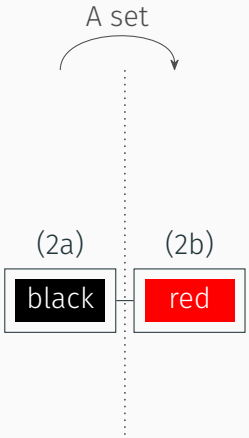


Figure 8: Canvas with one replicated register.

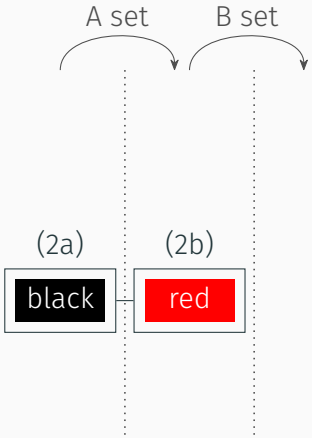


Figure 8: Canvas with one replicated register.

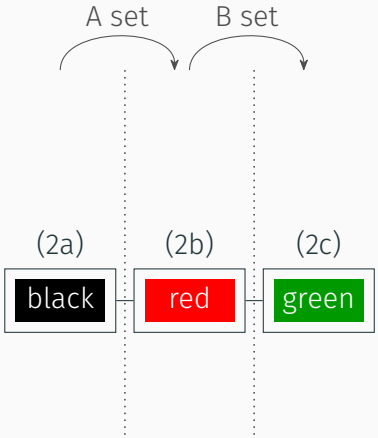


Figure 8: Canvas with one replicated register.

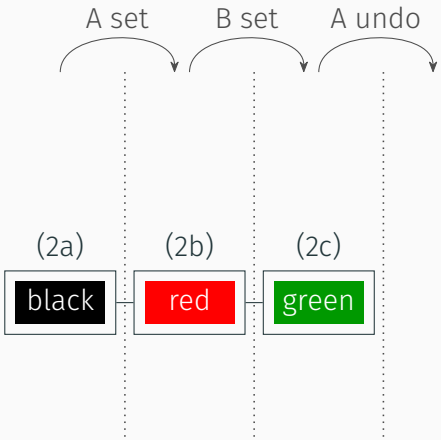


Figure 8: Canvas with one replicated register.

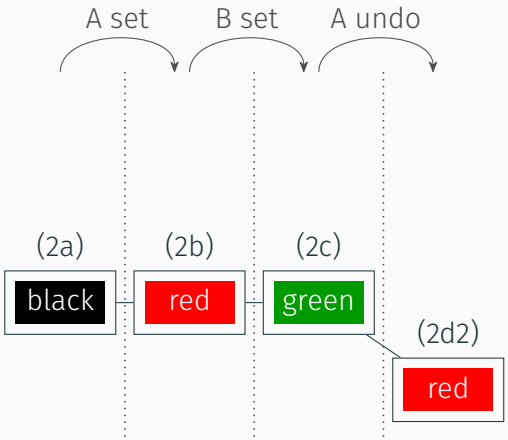


Figure 8: Canvas with one replicated register.

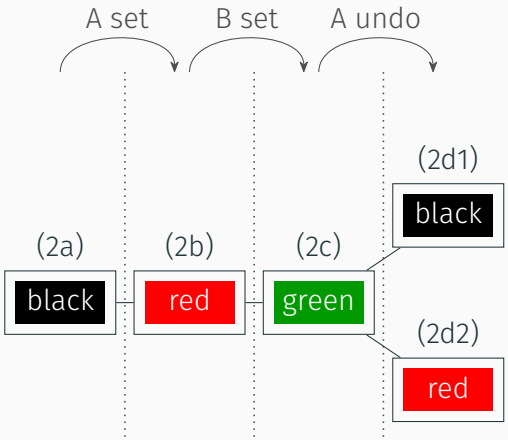


Figure 8: Canvas with one replicated register.

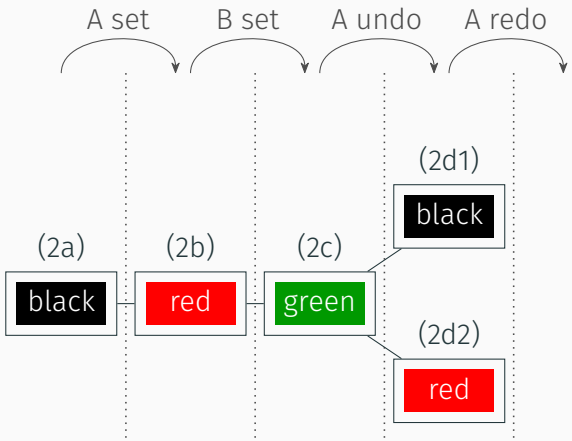


Figure 8: Canvas with one replicated register.

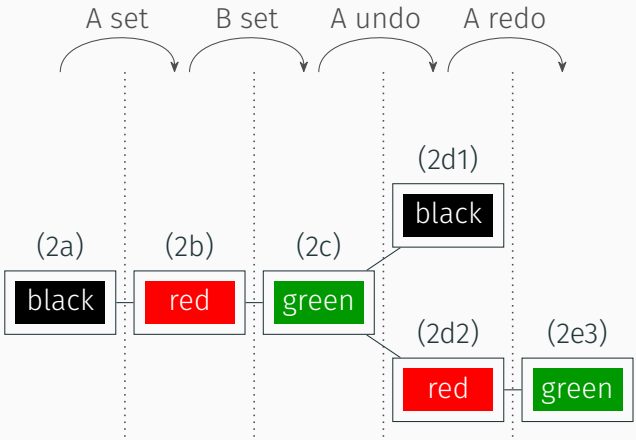


Figure 8: Canvas with one replicated register.

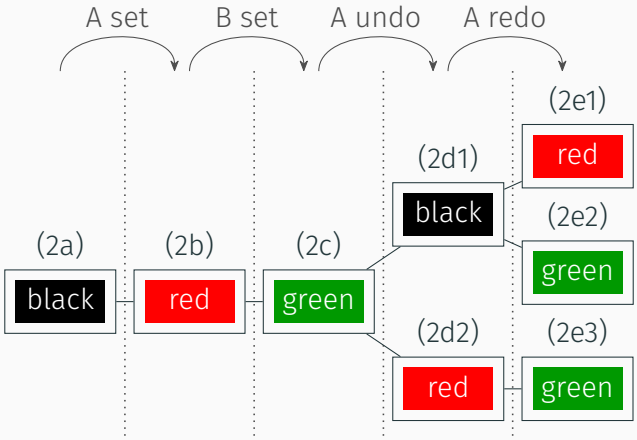


Figure 8: Canvas with one replicated register.

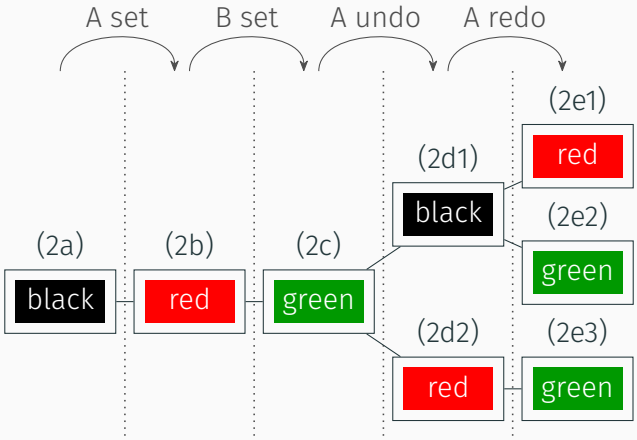


Figure 9: Canvas with one replicated register.

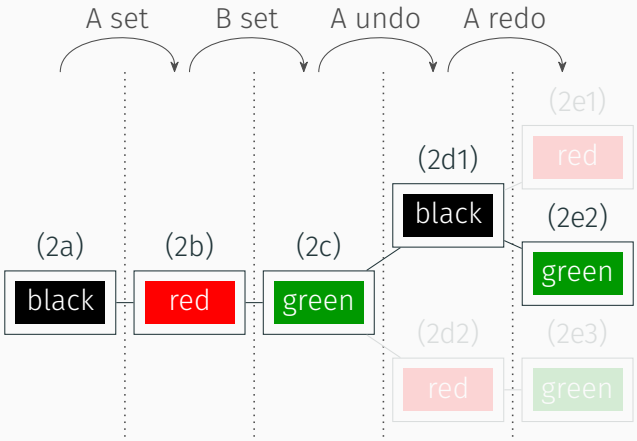


Figure 9: Canvas with one replicated register.

Register: [1]

Figure 10: An operation history of a single register with undo and redo.

Register: [2]

Figure 10: An operation history of a single register with undo and redo.

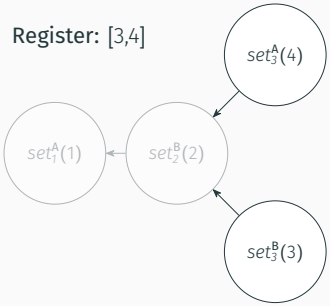


Figure 10: An operation history of a single register with undo and redo.

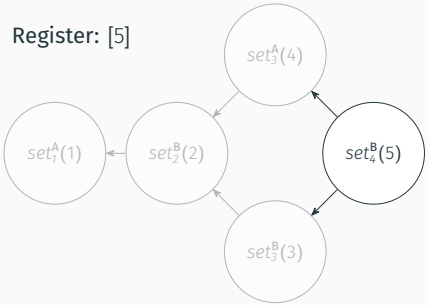


Figure 10: An operation history of a single register with undo and redo.

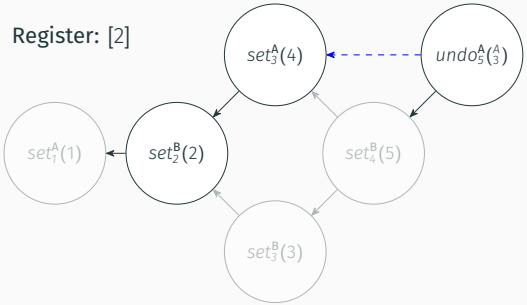


Figure 10: An operation history of a single register with undo and redo.

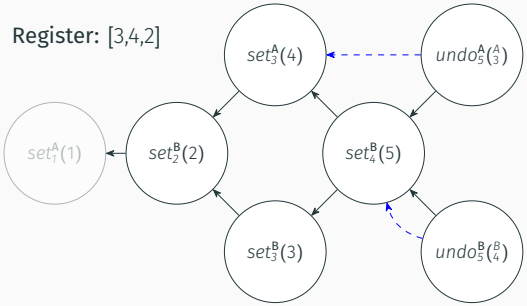


Figure 10: An operation history of a single register with undo and redo (backlink).

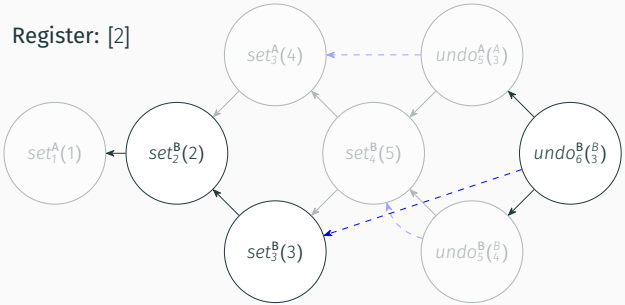


Figure 10: An operation history of a single register with undo and redo.

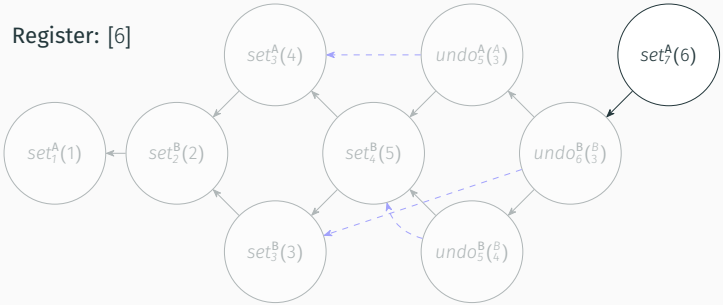


Figure 10: An operation history of a single register with undo and redo.

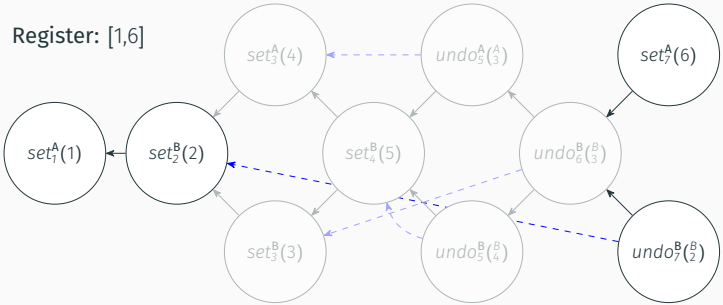


Figure 10: An operation history of a single register with undo and redo.

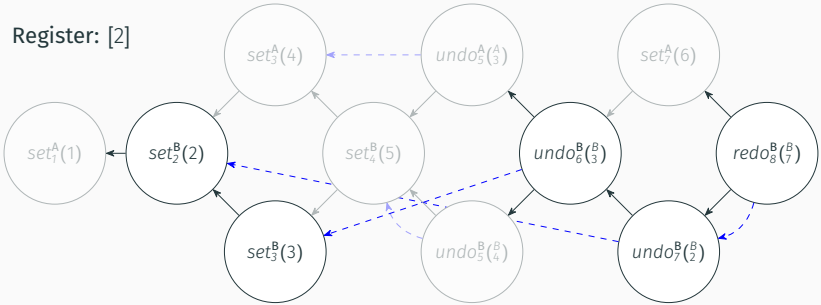


Figure 10: An operation history of a single register with undo and redo.

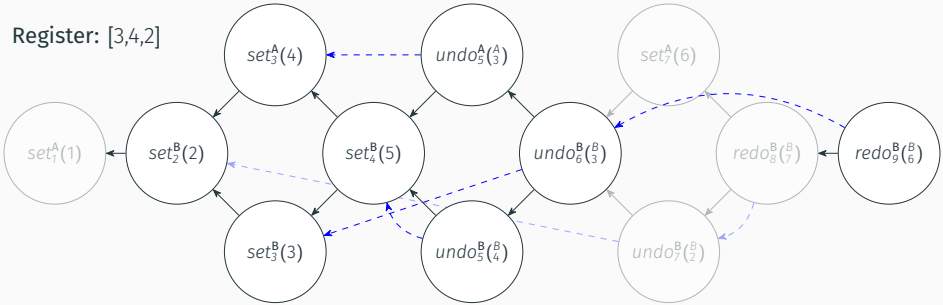


Figure 10: An operation history of a single register with undo and redo (to undo).

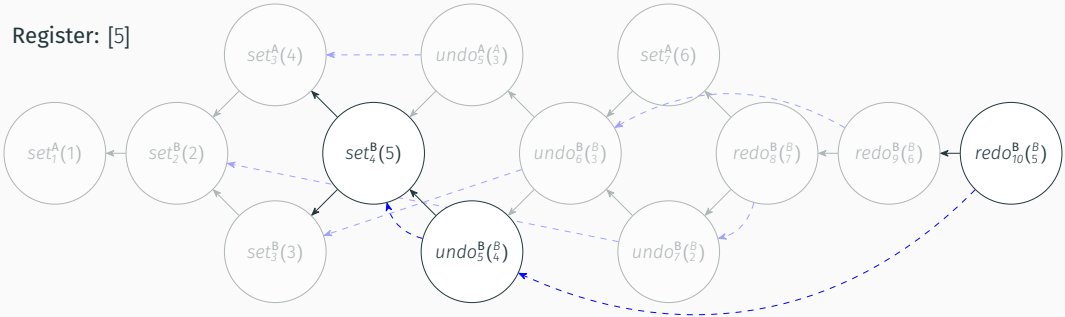


Figure 10: An operation history of a single register with undo and redo.

Degenerate Editing Scenario

Figure 11: Sequence of alternating undo-redo operations.

Degenerate Editing Scenario

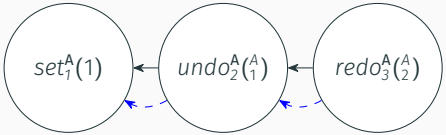


Figure 11: Sequence of alternating undo-redo operations of length 1.

Degenerate Editing Scenario

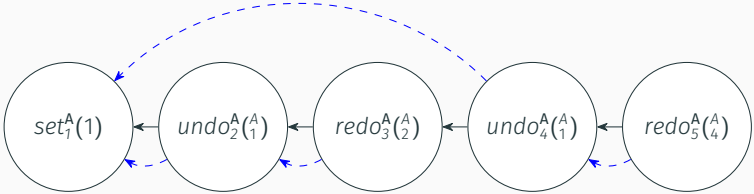


Figure 11: Sequence of alternating undo-redo operations of length 2.

Degenerate Editing Scenario

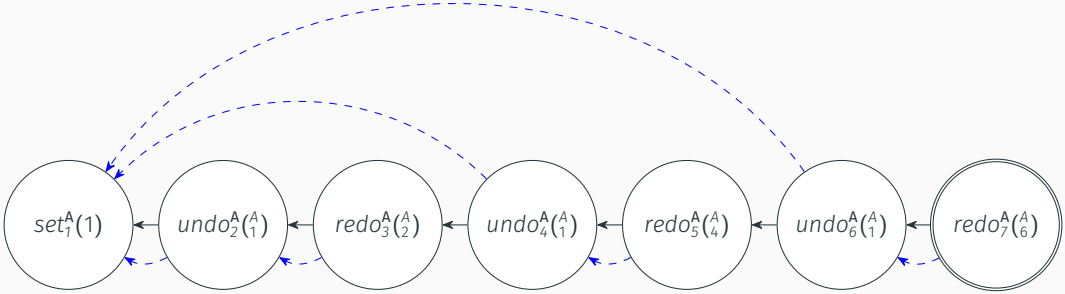


Figure 11: Sequence of alternating undo-redo operations of length 3.

Linear Runtime for Impractical Scenarios

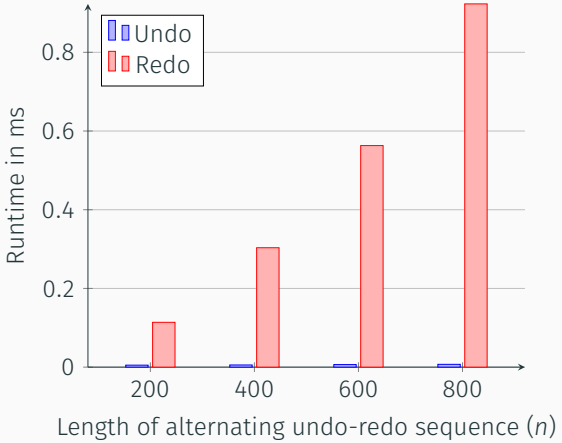


Figure 12: Runtime of the last undo/redo operation in a sequence of alternating undo-redo operations of length n.

A Taxonomy of Undo Behavior

Order \ Origin	Generating Replica	Any Replica
Reverse Chronological	local undo	global undo
Selective	revert[1]	

[1] often called *selective undo* in the literature [1]